

GIT Tutorial

1. Introduction:

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git allows group of people to work on the same project at the same time.

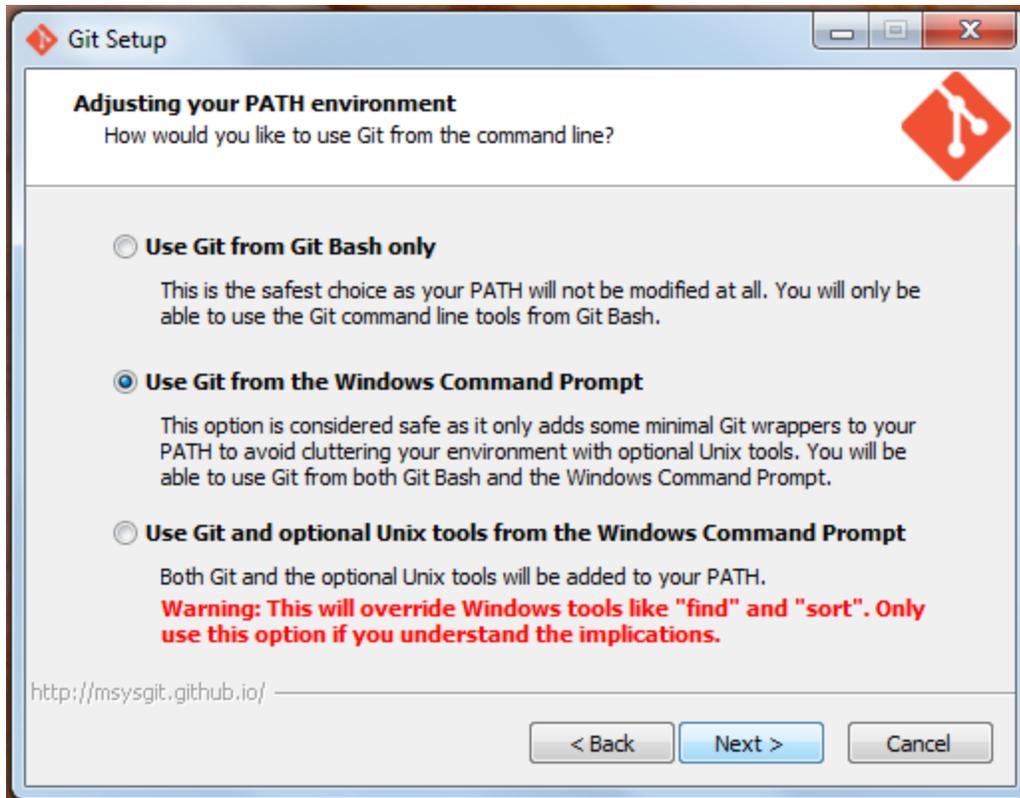
2. Getting Started

2.1 Installation of Git on Windows

- i. Go to link <http://git-scm.com/download/win> and download the Git package for windows.
- ii. Run the Package.



- iii. Install by default setting and click next. Make sure you select second option as shown below. It will allow you to run git command from command window.



After installing the package you will have “Git GUI” and “Git Bash” applications on your computer.

2.2 Authentication with GitHub

i. Configure your name and email address using following commands. This information will be associated with your commits.

```
$ git config --global user.email "Your Email Address"  
$ git config --global user.name "Your Name"
```

ii. Create Public and Private Key.

```
$ ssh-keygen -t rsa -C "yourid@stevens.edu"
```

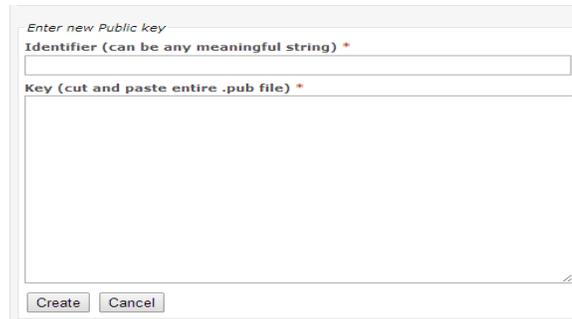
Accept the default values. You do not need to create passphrase.

iii. Check your key at location specified during key generation. If you do not specify any location, by default the files will be stored at c:\Users\username\.ssh

Two files will be created in this directory with default names - id_rsa and id_rsa.pub.

iv. Open the pub file and copy everything from the file

v. Log in to Redmine <https://kolmogorov.fsc.stevens.edu/redmine>. Click on “My account” and paste the copied key from the pub file in “key” field. Give the name to the key in “Identifier” field. And click on Create.



2.3 Clone the existing Git Repository.

In Windows, there are two options to clone the Existing Git repository.

2.3.1 Using Git Bash Application

i. Go to your project on Redmine. Click on the Repository Tab. Clone the Git repository using the “Repository access link” provided under ssh tab.



ii. Use the “git clone” command in “Git Bash” application to clone the repository.

```
$ git clone git@kolmogorov.fsc.stevens.edu:client.git
```

2.3.2 Using Git GUI Application

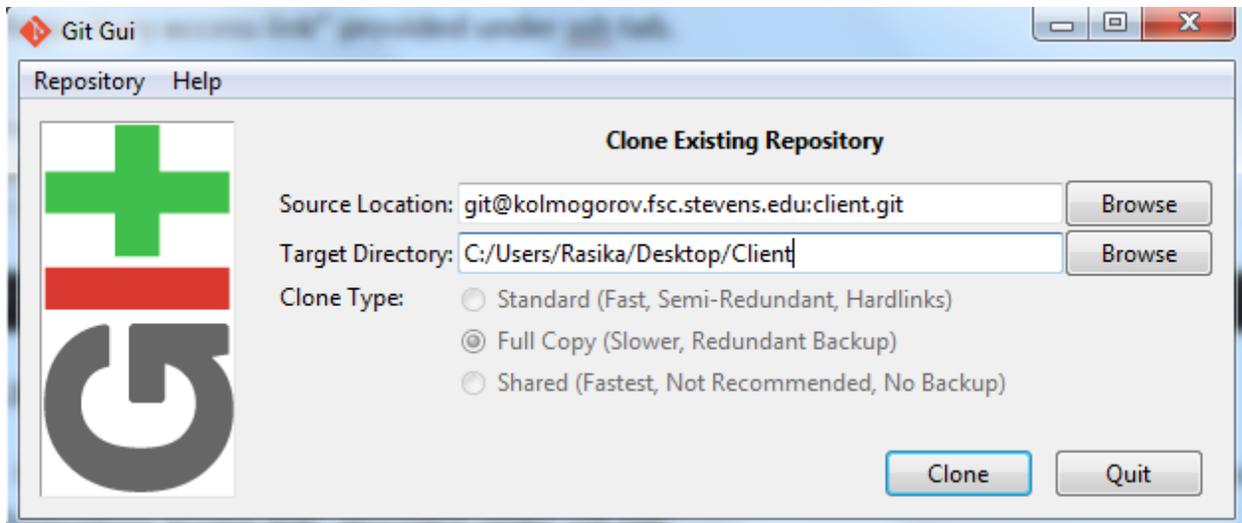
i. Go to your project in Redmine. Click on the Repository Tab. Copy the link provided in “Repository access link” under ssh tab.

ii. Open “Git GUI” Application. Click on “Clone Existing Repository”

iii. Paste the “Repository access link in “Source Location” and Specify the Target Directory.

Note: Give the Non existing Folder Name in “Target Directory”. The folder with specified name will be created while cloning the repository.

iii. Click on “Clone”



2.4 Update the code into the git repository.

2.4.1 Using Git bash : Follow the steps given below to upload the code without any conflict in the git repository.

i. Before adding files always update the git repository using command 'git pull' at the location where git repository is cloned.

```
$ git pull_
```

ii. Now add all the files where you made the changes using 'git add -A' command.

```
$ git add -A_
```

iii. Once all the files are added. Commit the files by adding appropriate description mentioning the changes by using 'git commit -m "Add comment"'

```
$ git commit -m "Add funcion to connect to database"
```

iv. Finally push the code into the git repository by using git push command.

```
$ git push_
```

3. Git for Linux

3.1 Installation of Git on Linux

- Use the following link and choose the appropriate command based on the Linux distribution that you are using.

<http://git-scm.com/download/linux>

- Generate your public and private key

```
$ ssh-keygen -t rsa -C "youremail@yourcompany.edu"
```

Accept the default (push enter every time you are asked) - do not create a passphrase - this is not needed and only used to store the key on your machine. The key is located at the directory specified when you generated the key (On Linux it is in .ssh directory). There will be two files in that directory. (- id_rsa and id_rsa.pub if you used default names).

- Open the public key file with any editor (e.g. gedit).
- Copy everything in the file and upload your public key into the following link.

<https://kolmogorov.fsc.stevens.edu/redmine/my/account>

Note: You need to log in to <https://kolmogorov.fsc.stevens.edu/redmine>

- Clone the repository

In redmine go to the project you are working on, click on repository, under the SSH line you see the command you need to run in your git BASH.

(e.g. for ACTUS \$ git clone [git@kolmogorov.fsc.stevens.edu](https://kolmogorov.fsc.stevens.edu):actus-master.git)

- Set up your identity by following commands

```
$ git config --global user.name "Mike Ross"
```

```
$ git config --global user.email "mross@example.com"
```

3.2 Working with Git in Linux

- Create a directory

```
$ mkdir <directory name>
```

- Go to that directory

```
$ cd <directory name>
```

- Initialize a Git repository

```
$ git init
```

- Check the current status of projects

```
$ git status
```

- create a file like ' test.txt'

\$ touch test.txt

- Again, use 'git status' command. You will see that 'test.txt' is a untracked file, which means that Git sees that 'test.txt' is a new file.
- To tell Git to start tracking changes made to 'test.txt', we first need to add it to the staging area by using following command.

\$ git add test.txt

- Now, again check status of file by using 'git status' command. You will see that Git is now tracking 'test.txt'
- At this point, the 'test.txt' file is in the staging area, and it is not in our repository yet. We could add or remove files from the stage before we store them in the repository. To store our changes, use following command with a message describing what we have changed.

\$ git commit -m 'This is initial commit'

- After making commit, we can browse them to see what we changed. By using 'git log' command, we will see all the changes that have been committed so far in the order we committed them.
- To push our local repository to the Git server you need to add a remote repository and then push the files.

\$git remote add origin <URL from my account in redmine>

\$ git push -u origin master

Note: The name of our remote is origin and the default local branch is master. The -u tells Git to remember the parameters, so that next time we can simply run git push and Git will know what to do

- Let's say some people have pulled our changes, made their own commits, and pushed them. We can check for changes on our Git repository and pull down any new changes.

\$ git pull origin master

- If there have been some changes to the files, we can take a look at what is different from our last commit by using the following command.

\$ git diff HEAD

More Information can be found in the following web sites

- <https://www.atlassian.com/git/tutorial>
- <http://www.vogella.com/tutorials/Git/article.html>
- <http://lostechies.com/joshuaflanagan/2010/09/03/use-gitk-to-understand-git/>
- <https://try.github.io/levels/1/challenges/1>
- <http://www.sbf5.com/~cduan/technical/git/git-1.shtml>